CLAIMS

I claim:

1. A method for selectively emulating sequential consistency in software comprising:
(a) forcing each CPU to execute a memory barrier instruction; and
(b) having each CPU send an indicator communicating completion of said memory barrier instruction.

2. The method of claim 1, wherein the step of forcing each CPU to execute a memory barrier instruction includes sending an interprocessor interrupt to all CPUs.

3. The method of claim 2, further comprising avoiding deadlock by having each CPU that is waiting for other CPUs to execute said memory barrier instruction continuously satisfy concurrent memory barrier execution requests.

4. The method of claim 1, further comprising using a single set of memory barrier instruction executions to satisfy groups of concurrent memory barrier execution requests.

5. The method of claim 4, further comprising assigning a generation number to each group of said memory barrier execution requests.

6. The method of claim 5, further comprising assigning a current generation number to all arriving memory barrier execution requests while a previous memory barrier execution request is being serviced.

7. The method of claim 1, wherein the step of having each CPU send an indicator communicating completion of said memory barrier instruction comprises said CPUs registering completion of said memory barrier with a tool selected from the group consisting of: an array, a bitmask, and a combining tree.

8. The method of claim 1, further comprising providing an array for CPUs to register memory barrier requests, wherein each array entry corresponds to a memory barrier execution request from a CPU.

9. The method of claim 8, further comprising scanning said array to determine execution of said memory barrier instruction by each CPU.

10. A computer system, comprising:
multiple processors;
an instruction for forcing each CPU to execute a memory barrier instruction; and
an instruction manager for indicating completion of said memory barrier instruction.

11. The system of claim 10, wherein the instruction for forcing each CPU to execute a memory barrier instruction comprises a memory barrier manager for sending an interprocessor interrupt to all CPUs.

12. The system of claim 11, wherein the memory barrier manager includes a waiting instruction to require each CPU waiting for other CPUs to execute said memory barrier instructions to satisfy concurrent memory barrier execution requests.

13. The system of claim 10, wherein the instruction for forcing each CPU to execute a memory barrier instruction comprises a consolidation instruction to satisfy groups of concurrent memory barrier execution requests with a single set of memory barrier instruction executions.

14. The system of claim 13, wherein each group of said memory barrier execution requests is assigned a generation number.

15. The system of claim 14, wherein all memory barrier execution requests that arrive while a previous memory barrier execution request is executing are assigned current generation numbers.

16. The system of claim 10, wherein the instruction manager includes a tool for each CPU to register completion of said memory barrier instruction.

17. The system of claim 16, wherein said tool is selected from the group consisting of: a bitmask, an array, and a combining tree.

18. The system of claim 10, wherein the instruction manager includes array for CPUs to register memory barrier requests.

19. The system of claim 19, wherein each entry to said array corresponds to a memory barrier execution request from a CPU.

20. The system of claim 19, wherein each requesting CPU scans said array to determine execution of said memory barrier instruction by each CPU.

21. An article comprising:
a computer-readable signal bearing medium;
means in the medium for forcing each CPU to execute a memory barrier instruction; and
an instruction manager for indicating completion of said memory barrier instruction

22. The article of claim 21, wherein the medium is selected from the group consisting of: a recordable data storage medium, and a modulated carrier signal.

23. The article of claim 21, wherein the means for forcing each CPU to execute the memory barrier instruction includes a memory barrier manager for sending an interprocessor interrupt to all CPUs.

24. The article of claim 21, wherein each CPU waiting for other CPUs to execute said memory barrier instruction continuously satisfies concurrent memory barrier execution requests.

25. The article of claim 21, wherein the instruction manager includes a tool for each CPU to register completion of said memory barrier instruction.

26. The article of claim 25, wherein said tool is selected from the group consisting of: a bitmask, an array, and a combining tree.

27. A method for selectively emulating sequential consistency in software comprising:

(a) forcing each CPU to execute a memory barrier instruction;

(b) having each CPU send an indicator communicating completion of said memory barrier instruction;

(c) satisfying groups of concurrent memory barrier execution requests with a single set of memory barrier instruction executions; and

(d) wherein the step of forcing each CPU to execute a memory barrier instruction includes sending an interprocessor interrupt to all CPUs forcing execution of said memory barrier instruction.

28. The method of claim 27, wherein the step of having each CPU send an indicator communicating completion of said memory barrier instruction comprises said CPUs registering completion of said memory barrier with a tool selected from the group consisting of: a bitmask, an array, and a combining tree.